PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-266554

(43)Date of publication of application : 22.09.1994

(51)Int.Cl. G06F 9/30
G06F 7/00
G06F 15/72

(21)Application number : 05-056792

(22)Date of filing : 17.03.1993

(71)Applicant : HITACHI LTD

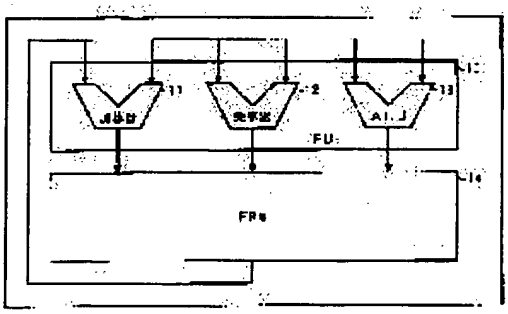
(72)Inventor : SUZUKI KATSUNORI
FUJITA MAKOTO
KOGA KAZUYOSHI
FUJII HIDEKI

(54) MICROPROCESSOR

(57)Abstract:

PURPOSE: To provide a microprocessor suitable for a graphic system capable of remarkably accelerating a packing processing and an unpacking processing and to provide the graphic system to which the microprocessor is applied.

CONSTITUTION: This microprocessor for performing the geometrical processing of the graphic system is provided with a floating register, an adder 11, a multiplier 12 and an ALU 13 for performing a bit arithmetic processing inside a floating (floating point real number) arithmetic unit. Thus, since the microprocessor can perform the bit arithmetic processing with the floating register, the packing processing and the unpacking processing can be performed with the floating register at a high speed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of

rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平6－266554

(43)公開日　平成６年(1994)９月22日

(51)Int.Cl.[5]		識別記号	庁内整理番号	Ｆ Ｉ		技術表示箇所
Ｇ０６Ｆ	9/30	３５０　Ｅ	9189－5B			
	7/00					
	15/72					
			9188－5B	Ｇ０６Ｆ	7/00　１０１　Ｗ	

審査請求　未請求　請求項の数４　ＯＬ　（全 13 頁）

(21)出願番号　特願平5－56792

(22)出願日　平成５年(1993)３月17日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　鈴木　克徳
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内

(72)発明者　藤田　良
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内

(72)発明者　古賀　和義
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内

(74)代理人　弁理士　小川　勝男

最終頁に続く

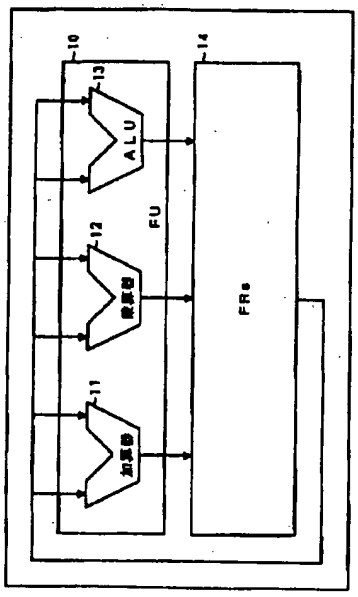
(54)【発明の名称】　マイクロプロセッサ

(57)【要約】

【目的】本発明は、pack処理，pnpack処理を著しく高速化することができるグラフイックス・システムに適したマイクロプロセッサ、及びそれを適用したグラフイックス・システムを提供することを目的としている。

【構成】前記目的を達成するために、本発明のグラフィックス・システムの幾何処理を行うマイクロプロセッサは、floating（浮動小数点実数）演算ユニット内に、floatingレジスタと、加算器と、乗算器と、bit 演算処理を行うＡＬＵを有する。これにより、本発明のマイクロプロセッサは、floatingレジスタ間でbit 演算処理を行うことができるので、pack処理，unpack処理をfloatingレジスタ間で高速に行うことが可能である。

図　1



1

【特許請求の範囲】

【請求項１】グラフィックス・システムの幾何処理を行うマイクロプロセッサにおいて、

floating（浮動小数点実数）演算ユニット内に、floatingレジスタと、加算器と、乗算器と、bit演算処理を行うＡＬＵを有し、（１）floatingレジスタにある１画素のデータを構成する各成分を示す複数のfloating型のデータを加算器によりfloating型からinteger(整数）型へ変換し、（２）前記変換結果である各成分を示す複数のinteger型のデータの有効なbit部分をＡＬＵにより１つの画素データに変換するpack処理をfloatingレジスタ間で行うことを特徴とするマイクロプロセッサ。

【請求項２】グラフィックス・システムの幾何処理を行うマイクロプロセッサにおいて、

floating（浮動小数点実数）演算ユニット内に、floatingレジスタと、加算器と、乗算器と、bit 演算処理を行うＡＬＵを有し、（１）floatingレジスタにある複数の成分から構成される１つの画素データをＡＬＵにより各成分を示す複数のinteger 型のデータに変換し、（２）前記変換結果である各成分を示す複数のinteger 型のデータを加算器によりinteger 型からfloating型へ変換するunpack処理をfloatingレジスタ間で行うことを特徴とするマイクロプロセッサ。

【請求項３】グラフィックス・システムの幾何処理を行うマイクロプロセッサにおいて、

floating（浮動小数点実数）演算ユニット内に、floatingレジスタと、加算器と、乗算器と、bit 演算処理を行うＡＬＵを有し、（１）floatingレジスタにある１画素のデータを構成する各成分を示す複数のfloating型のデータを加算器によりfloating型からinteger （整数）型へ変換し、（２）前記変換結果である各成分を示す複数のinteger 型のデータの有効なbit部分をＡＬＵにより１つの画素データに変換するpack処理をfloatingレジスタ間で行う命令を有することを特徴とするマイクロプロセッサ。

【請求項４】グラフィックス・システムの幾何処理を行うマイクロプロセッサにおいて、

floating（浮動小数点実数）演算ユニット内に、floatingレジスタと、加算器と、乗算器と、bit 演算処理を行うＡＬＵを有し、（１）floatingレジスタにある複数の成分から構成される１つの画素データをＡＬＵにより各成分を示す複数のinteger 型のデータに変換し、（２）前記変換結果である各成分を示す複数のinteger 型のデータを加算器によりinteger 型からfloating型へ変換するunpack処理をfloatingレジスタ間で行う命令を有することを特徴とするマイクロプロセッサ。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、グラフィックス・システムに適したマイクロプロセッサ、及びそれを適用したグラフィックス・システムに関する。更に詳細にいえば、グラフィックス・システムにおける画素データのpack処理とunpack処理、即ち、（１）１画素のデータを構成する各成分を示す複数のfloating型のデータをinteger 型へ変換し、前記変換結果である各成分を示す複数のデータの有効なbit 部分を１つの画素データに変換する処理と、（２）複数の成分から構成される１つの画素データを各成分を示す複数のデータに変換するbit 演算処理と、前記変換結果である各成分を示す複数のinteger 型のデータをfloating型へ変換する処理の高速化に関する。

【０００２】

【従来の技術】コンピュータ・グラフィックス・システムは、コンピュータの出力を図形として表示するものである。従来のグラフィックス・システムの図形表示方法は、「PEX Introduction and Overview (M.I.T., 1988, pp51－72)」に詳細が記載されている。

【０００３】グラフィックス・システムは、図形の基準点の座標変換と、図形，光源，視点の位置，色などの情報より図形がどの様に見えるか光源計算を行い図形の基準点の色を算出する幾何処理と、図形の基準点の情報からその図形の内部の画素を内挿補間により１画素ずつ展開して描画するレンダリング処理を行い、ディスプレイに表示する内容をビットマップ形式で保持する画像メモリに書き込む。グラフィックス・システムは、一般に、画素データを図形の座標データＸＹＺと色データＲＧＢ(Red，Green，Blue)で表現し、前記幾何処理ではfloating（浮動小数点実数）型で計算し、前記レンダリング処理ではinteger （整数）型で計算する。ここでは、説明を簡単にするために、画像メモリの画素データの色データＲＧＢの各成分が各８bit 、合計２４bit であるグラフィックス・システムについて説明する。

【０００４】前記の様に、floating型のｒ，ｇ，ｂをinteger 型のＩｒ，Ｉｇ，Ｉｂに変換し、各成分別々なＩｒ，Ｉｇ，Ｉｂの有効な各８bit を１つの２４bit のデータに変換して画像メモリの画素データを表現するため、グラフィックス・システムでは、ＲＧＢ各成分を示す複数のfloating型のデータｒ，ｇ，ｂを、ＲＧＢ各成分を示す複数のinteger 型のデータＩｒ，Ｉｇ，Ｉｂに変換するデータ型の変換処理と、前記変換結果であるＩｒ，Ｉｇ，Ｉｂの有効な各８bit 部分を１つの画素データに変換するbit 演算処理が画像メモリ書き込み時に必ず発生する。ここでは、この一連の処理をpack処理と呼ぶ。又、その逆変換処理、即ち、１つの画素データをＲＧＢ各成分を示す複数のinteger 型のデータＩｒ，Ｉｇ，Ｉｂに変換するbit 演算処理を行い、前記変換結果であるＩｒ，Ｉｇ，ＩｂをＲＧＢ各成分を示す複数のfloating型のデータｒ，ｇ，ｂに変換するデータ型の変換処理も発生する。ここでは、この一連の処理をunpack処理と呼ぶ。

【０００５】次に、従来のpack処理とunpack処理について説明する。

【０００６】まず、従来のpack処理について説明する。

【０００７】図４に、従来のpack処理のアセンブリ言語によるプログラムとレジスタの状態の例を示す。

【０００８】但し、％grはinteger 演算を行うためのgeneral（汎用）レジスタ，％frはfloating演算を行うためのfloatingレジスタ，／＊…＊／はコメント（注釈）である。

【０００９】（１）初めに、Ｌ４０２，４０３，４０４では、初期状態として、ＲＧＢの各成分を示す複数のfloating型のデータｒ，ｇ，ｂがfloatingレジスタ１６，１７，１８番にある（４０２，４０３，４０４）。

【００１０】（ｒ，ｇ，ｂはfloating型，０.０≦ｒ，ｇ，ｂ≦２５５.０）

（２）そして、Ｌ４０６，４０７，４０８では、floating型のデータをinteger型のデータに変換する命令fcnvfxにより、前記データｒ，ｇ，ｂをinteger 型のデータＩｒ，Ｉｇ，Ｉｂへ変換するデータ型の変換処理を順次３回行う（４０６，４０７，４０８）。

【００１１】（Ｉｒ，Ｉｇ，Ｉｂはinteger 型，０≦Ｉｒ，Ｉｇ，Ｉｂ≦２５５）

（３）次に、bit 演算処理を行うわけだが、一般にfloatingレジスタではbit 演算処理を行うことができない。そこで、bit 演算処理を行うために、Ｌ４１０，４１１，４１２では、floatingレジスタからgeneral レジスタへデータを転送する命令frtogrにより、前記データＩｒ，Ｉｇ，Ｉｂをfloatingレジスタ１６，１７，１８番からbit 演算処理が可能なgeneral レジスタ１６，１７，１８番へ転送するデータ転送処理を順次３回行う（４１０，４１１，４１２）。

（４）最後に、前記データＩｒ，Ｉｇ，Ｉｂの有効な各８bitを１つの画素データに変換するために、Ｌ４１４，４１５，４１６ではbit 演算処理を行う命令dep により、general レジスタ１６，１７，１８番にある前記Ｉｒ，Ｉｇ，Ｉｂの有効な各８bit をgeneral レジスタ１９番の各８bit の部分に設定するbit 演算処理を順次３回行う（４１４，４１５，４１６）。

【００１２】前記の様に従来のpack処理では、floatingレジスタでfloating型のデータｒ，ｇ，ｂをinteger 型のデータＩｒ，Ｉｇ，Ｉｂへ変換した後、floatingレジスタでは一般にbit 演算処理を行えないので、floatingレジスタからbit 演算処理が可能なgeneral レジスタへ転送し、general レジスタでbit 演算処理を行い１つの画素データに変換する。このため、floatingレジスタからgeneral レジスタへのデータ転送処理と、更に、順次３回のbit 演算処理が必要となり、pack処理に多大な時間を要するという問題を有する。

【００１３】次に、従来のunpack処理について説明する。

【００１４】図５に、従来のunpack処理のアセンブリ言語によるプログラムとレジスタの状態の例を示す。

【００１５】（１）初めに、Ｌ５０２では、初期状態として、ＲＧＢの各成分Ｉｒ，Ｉｇ，Ｉｂの各８bit を既にpack処理した画素データがgeneral レジスタ１９番にある(５０２)。

【００１６】（Ｉｒ，Ｉｇ，Ｉｂはinteger 型，０≦Ｉｒ，Ｉｇ，Ｉｂ≦２５５）

（２）そして、Ｌ５０４，５０５，５０６ではbit 演算処理を行う命令extru により、前記Ｉｒ，Ｉｇ，Ｉｂ各８bit をそれぞれgeneral レジスタ16，17，18番に設定するbit 演算処理を順次３回行う（５０４，５０５，５０６）。

（３）次に、データ型の変換処理を行うわけだが、一般にgeneral レジスタではデータ型の変換処理を行うことができない。そこで、データ型の変換処理を行うために、Ｌ５０８，５０９，５１０ではgeneral レジスタからfloatingレジスタへデータを転送する命令grtofrにより、前記データＩｒ，Ｉｇ，Ｉｂをgeneral レジスタ１６，１７，１８番からデータ型の変換処理が可能なfloatingレジスタ１６，１７，１８番へ転送するデータ転送処理を順次３回行う（５０８，５０９，５１０）。

【００１７】（４）最後に、Ｌ５１２，５１３，５１４ではinteger 型のデータをfloating型のデータに変換する命令fcnvxfにより、floatingレジスタ１６，１７，１８番にある前記データＩｒ，Ｉｇ，Ｉｂをfloating型のデータｒ，ｇ，ｂへ変換するデータ型の変換処理を順次３回行う（５１２，５１３，５１４）。

【００１８】（ｒ，ｇ，ｂはfloating型，０.０≦ｒ，ｇ，ｂ≦２５５.０）

前記の様に従来のunpack処理では、１つの画素データをgeneralレジスタでbit演算処理した後、general レジスタからデータ型の変換処理が可能なfloatingレジスタへ転送し、floatingレジスタでinteger 型のデータをfloating型のデータへ変換する。このため、general レジスタからfloatingレジスタへのデータ転送処理と、更に、順次３回のbit 演算処理を必要となり、unpack処理に多大な時間を要するという問題を有する。

【００１９】

【発明が解決しようとする課題】前記の様に従来のpack処理では、floatingレジスタからgeneral レジスタへのデータ転送処理と、更に、順次３回のbit 演算処理を必要とし、pack処理に多大な時間を要するという問題を有する。

【００２０】又、従来のunpack処理では、general レジスタからfloatingレジスタへのデータ転送処理と、更に、順次３回のbit 演算処理を必要とし、unpack処理に多大な時間を要するという問題を有する。

【００２１】従って、本発明は、前記問題点を解決して、pack処理，unpack処理を著しく高速化することがで

きるグラフィックス・システムに適したマイクロプロセッサ、及びそれを適用したグラフィックス・システムを提供することを目的としている。

【００２２】

【課題を解決するための手段】前記目的を達成するために、本発明のグラフィックス・システムの幾何処理を行うマイクロプロセッサは、floating（浮動小数点実数）演算ユニット内に、floatingレジスタと、加算器と、乗算器と、bit 演算処理を行うＡＬＵを有する。これにより、本発明のマイクロプロセッサは、floatingレジスタ間でbit 演算処理を行うことができるので、floatingレジスタ間でpack処理，unpack処理を高速に行うことが可能となる。

【００２３】以上の様に構成すれば、グラフィックス・システムにおける画素データのpack処理，unpack処理を高速に行うことができるグラフィックス・システムに適したマイクロプロセッサを構成できる。又、それをグラフィックス・システムに適用することにより、画素データのpack処理，unpack処理を著しく高速に行うことができるグラフィックス・システムを構成できる。

【００２４】

【作用】本発明のグラフィックス・システムの幾何処理を行うマイクロプロセッサは、pack処理において、floatingレジスタにあるfloating型のデータを加算器によりinteger 型のデータへ変換し、前記変換結果であるinteger 型のデータの有効なbit 部分をＡＬＵによりそのままfloatingレジスタでbit 演算処理を行い１つの画素データに変換する。この様に、本発明のマイクロプロセッサは、floatingレジスタからgeneral レジスタへのデータ転送処理を行わないので、pack処理を著しく高速化することが可能となる。

【００２５】又、本発明のグラフィックス・システムの幾何処理を行うマイクロプロセッサは、unpack処理において、floatingレジスタにある複数の成分から構成される１つの画素データをＡＬＵによりそのままfloatingレジスタで各成分を示す複数のinteger 型のデータに変換するbit演算処理を行い、前記変換結果であるinteger型のデータを加算器によりfloating型のデータへ変換する。この様に、本発明のマイクロプロセッサは、general レジスタからfloatingレジスタへのデータ転送処理を行わないので、unpack処理を著しく高速化することが可能となる。

【００２６】以上の様に、本発明のマイクロプロセッサは、グラフィックス・システムにおける画素データのpack処理，unpack処理を著しく高速化することができる。又、それを適用したグラフィックス・システムは、グラフィックス・システムにおける画素データのpack処理，unpack処理を著しく高速化することが可能となる。

【００２７】

【実施例】以下、実施例を図面によって詳細に説明する。

【００２８】図１０は、本発明のグラフィックス・システムの一実施例を示す構成図である。

【００２９】本発明のグラフィックス・システムは、幾何処理を行うマイクロプロセッサＣＰＵ（１０１），メモリ（１０３）等を制御するコントローラ（１０２），レンダリング処理を行う描画プロセッサ（１０４），描画した画像を保持する画像メモリ（１０５），画像を表示するディスプレイ（１０６）から構成される。

【００３０】ＣＰＵ（１０１）は、アプリケーションを実行し幾何処理を行い図形の基準点の画素データとグラフィックス・コマンド（描画コマンド）を発行し、コントローラ（１０２）を通して描画プロセッサ（１０４）に送出する。描画プロセッサ（１０４）は、図形の基準点の画素データとグラフィックス・コマンドから図形の内部の画素を内挿補間により１画素ずつ展開して描画するレンダリング処理を行い、ディスプレイに表示する内容をビットマップ形式で保持する画像メモリ（１０５）に書き込み、画像をディスプレイ（１０６）に表示する。

【００３１】グラフィックス・システムは、画素データを、幾何処理ではfloating（浮動小数点実数）型で計算し、レンダリング処理ではinteger （整数）型で計算する。ここでは、説明を簡単にするために、画像メモリの画素データの色データＲＧＢの各成分が各８bit，合計２４bit であるグラフィックス・システムについて説明する。

【００３２】まず、幾何処理を行うマイクロプロセッサＣＰＵ（１０１）について説明する。

【００３３】図２は、本発明のマイクロプロセッサの一実施例を示す構成図である。

【００３４】本発明のマイクロプロセッサは、マイクロプロセッサと外部を接続するSystem Interface（２１），命令キャッシュＩＣ（２２），データキャッシュＤＣ（２３），マイクロプロセッサの各ユニットを制御するControler（２４），integer 演算を行うためのgeneral レジスタＧＲｓ（２５），integer 演算を実際に行うinteger 演算ユニットＩＵ（２６），floating演算を行うためのfloatingレジスタＦＲｓ（２７），floating演算を実際に行うfloating演算ユニットＦＵ（２８）から構成される。

【００３５】本発明のマイクロプロセッサは、System Interface（２１）を介して、命令とデータを読み込み、命令キャッシュＩＣ（２２），データキャッシュＤＣ（２３）に格納し、命令に従ってControler がマイクロプロセッサの各ユニットを制御し、必要なデータがＧＲｓ（２５），ＦＲｓ（２７）に格納され、ＩＵ（２６），ＦＵ（２８）が命令の処理を実行する。

【００３６】次に、ＦＵ（２８）の構成について詳細に説明する。

【００３７】図１は、本発明のマイクロプロセッサのＦＵとＦＲｓの一実施例を示す構成図である。

【００３８】ＦＵ（１０）は、加減算やデータ型の変換などの演算を行う加算器（１１），乗除算や平方根などの演算を行う乗算器（１２），pack処理，unpack処理で必要なbit 演算処理を行うＡＬＵ（１３）から構成される。ＦＵ（１０）は、命令に従って、ＦＲｓ（１４）のデータを読み込み、加算器（１１），乗算器（１２），ＡＬＵ（１３）で演算を行い、その結果のデータをＦＲｓ（１４）に書き込む。次に、本発明のマイクロプロセッサにおける高速なpack処理とunpack処理について詳細に説明する。

【００３９】まず、pack処理について説明する。

【００４０】図６に、本発明のマイクロプロセッサにおける高速なpack処理のアセンブリ言語によるプログラムとレジスタの状態の例を示す。

【００４１】（１）初めに、Ｌ６０２，６０３，６０４では、初期状態として、ＲＧＢの各成分を示す複数のfloating型のデータｒ，ｇ，ｂがfloatingレジスタ１６，１７，１８番にある（６０２，６０３，６０４）。

【００４２】（ｒ，ｇ，ｂはfloating型，０.０≦ｒ，ｇ，ｂ≦２５５.０）

（２）次に、Ｌ６０７では、floating型のデータをinteger 型のデータへ変換してbit 演算処理を行う命令fcnvfxdep により、前記floatingレジスタ１６番にあるデータｒをinteger 型のデータＩｒへ変換するデータ型の変換処理を前記加算器（１１）で行い、更に、前記データＩｒの有効な８bit をfloatingレジスタ１９番の所定の８bit の部分に設定するbit 演算処理を前記ＡＬＵ(１３）で行う（６０７）。同様に、Ｌ６０８，６０９では、前記データｇ，ｂについて同様な処理を行う（６０８，６０９）。この様にして、ｆｌｏａｔｉｎｇ型のデータｒ，ｇ，ｂをinteger 型のデータＩｒ，Ｉｇ，Ｉｂに変換し、integer 型のデータＩｒ，Ｉｇ，Ｉｂの有効な各８bit を１つの画素データに変換する。

【００４３】（Ｉｒ，Ｉｇ，Ｉｂはinteger 型，０≦Ｉｒ，Ｉｇ，Ｉｂ≦２５５）

前記の様に本発明のマイクロプロセッサにおける高速なpack処理では、floatingレジスタでfloating型のデータをinteger 型のデータへ変換し、従来のpack処理では必要としていたfloatingレジスタからgeneral レジスタへのデータ転送処理を行わずに、そのままfloatingレジスタでbit 演算処理を行い画素データに変換する。これにより、floatingレジスタからgeneral レジスタへのデータ転送処理が不要となり、pack処理を著しく高速化することができる。

【００４４】次に、unpack処理について説明する。

【００４５】図７に、本発明のマイクロプロセッサにおける高速なunpack処理のアセンブリ言語によるプログラムとレジスタの状態の例を示す。

【００４６】（１）初めに、Ｌ７０２では、初期状態として、ＲＧＢの各成分Ｉｒ，Ｉｇ，Ｉｂ各８bit を既にpack処理した画素データがfloatingレジスタ１９番にある（７０２）。

【００４７】（Ｉｒ，Ｉｇ，Ｉｂはinteger 型，０≦Ｉｒ，Ｉｇ，Ｉｂ≦２５５）

（２）次に、Ｌ７０５では、bit演算処理を行ってinteger型のデータをfloating型のデータへ変換する命令fextrucnvxfにより、前記floating レジスタ１９番にある画素データのデータＩｒの８bit をfloatingレジスタ１６番に設定するbit 演算処理を前記ＡＬＵ（１３）で行い、更に、前記データＩｒをfloating型のデータｒへ変換するデータ型の変換処理を前記加算器(１１)で行う(７０５)。同様に、Ｌ７０６，７０７では、前記データＩｇ，Ｉｂについて同様な処理を行う（７０６，７０７）。この様にして、前記画素データの各成分Ｉｒ，Ｉｇ，Ｉｂの各８bit を、各成分を示す複数のfloating型のデータｒ，ｇ，ｂへ変換する。

【００４８】（ｒ，ｇ，ｂはfloating型，０.０≦ｒ，ｇ，ｂ≦２５５.０）

前記の様に本発明のマイクロプロセッサにおける高速なunpack処理では、従来のunpack処理では必要としていたgeneral レジスタからfloatingレジスタへのデータ転送処理を行わずに、そのままfloatingレジスタで画素データのbit 演算処理を行い、integer 型のデータをfloating型のデータへ変換する。これにより、general レジスタからfloatingレジスタへのデータ転送処理が不要となり、unpack処理を著しく高速化することができる。

【００４９】次に、別のＦＵ（２８）の構成について詳細に説明する。

【００５０】図１に示した本発明のマイクロプロセッサのＦＵ（１０）は、加算器（１１），乗算器（１２），ＡＬＵ（１３）を１組ずつ有しているが、図３に示したように複数組有しても良い。

【００５１】図３は、加算器，乗算器，ＡＬＵを３組有する本発明のマイクロプロセッサのＦＵとＦＲｓの一実施例を示す構成図である。

【００５２】ＦＵ（３１０）は、加減算やデータ型の変換などの演算を行う加算器（３０１，３０２，３０３），乗除算や平方根などの演算を行う乗算器（３０４，３０５，３０６），pack処理，unpack処理で必要なbit 演算処理を行うＡＬＵ（３０７，３０８，３０９）から構成される。ＦＵ（３１０）は、命令に従って、ＦＲｓ（３１１）のデータを読み込み、加算器（３０１，３０２，３０３），乗算器（３０４，３０５，３０６），ＡＬＵ（３０７，３０８，３０９）で演算を行い、その結果のデータをＦＲｓ（３１１）に書き込む。

【００５３】次に、本発明のマイクロプロセッサの別のＦＵの構成における高速なpack処理とunpack処理について詳細に説明する。

【００５４】図８に、本発明のマイクロプロセッサにおける高速なpack処理のアセンブリ言語によるプログラムとレジスタの状態の例を示す。

【００５５】（１）初めに、Ｌ８０２，８０３，８０４では、初期状態として、ＲＧＢの各成分を示す複数のfloating型のデータｒ，ｇ，ｂがfloatingレジスタ１６，１７，１８番にある（８０２，８０３，８０４）。

【００５６】（ｒ，ｇ，ｂはfloating型，０.０≦ｒ，ｇ，ｂ≦２５５.０）

（２）次に、Ｌ８０７では、３つのfloating型のデータをinteger型のデータへ変換してbit 演算処理を行う命令fcnvfxpack３により、前記floatingレジスタ１６，１７，１８番にあるデータｒ，ｇ，ｂをinteger 型のデータＩｒ，Ｉｇ，Ｉｂへ変換するデータ型の変換処理を前記加算器(３０１，３０２，３０３)で行い、更に、前記データＩｒ，Ｉｇ，Ｉｂの有効な各８bit をfloatingレジスタ１９番の所定の８bit の部分に設定するbit 演算処理を前記ＡＬＵ (307,３０８，３０９）で行う（６０７）。この様にして、floating型のデータｒ，ｇ，ｂをinteger 型のデータＩｒ，Ｉｇ，Ｉｂに変換し、integer 型のデータＩｒ，Ｉｇ，Ｉｂの有効な各８bit を１つの画素データに変換する。

【００５７】（Ｉｒ，Ｉｇ，Ｉｂはinteger 型，０≦Ｉｒ，Ｉｇ，Ｉｂ≦２５５）

前記の様に本発明のマイクロプロセッサにおける高速なpack処理では、floatingレジスタでfloating型のデータをinteger 型のデータへ変換し、従来のpack処理では必要としていたfloatingレジスタからgeneral レジスタへのデータ転送処理を行わずに、更に、前記実施例ではそのままfloatingレジスタで順次３回必要としていたbit 演算処理を１回のbit 演算処理で行い画素データに変換する。これにより、floatingレジスタからgeneral レジスタへのデータ転送処理が不要となり、更に、bit 演算処理を１回に削減しているので、pack処理を著しく高速化することができる。

【００５８】図９に、本発明のマイクロプロセッサにおける高速なunpack処理のアセンブリ言語によるプログラムとレジスタの状態の例を示す。

【００５９】（１）初めに、Ｌ９０２では、初期状態として、ＲＧＢの各成分Ｉｒ，Ｉｇ，Ｉｂ各８bit を既にpack処理した画素データがfloatingレジスタ１９番にある（９０２）。

【００６０】（Ｉｒ，Ｉｇ，Ｉｂはinteger 型，０≦Ｉｒ，Ｉｇ，Ｉｂ≦２５５）

（２）次に、Ｌ９０５では、３回分のbit 演算処理を行ってfloating型のデータをinteger 型のデータへ変換する命令funpack３cnvxfにより、前記floatingレジスタ１９番にある画素データのデータＩｒ，Ｉｇ，Ｉｂの各８bit をfloatingレジスタ１６，１７，１８番に設定するbit 演算処理を前記ＡＬＵ（３０７，３０８，３０９）で行い、更に、前記データＩｒ，Ｉｇ，Ｉｂをfloating型のデータｒ，ｇ，ｂへ変換するデータ型の変換処理を前記加算器（３０１，３０２，３０３）で行う（９０６，９０７，９０８）。この様にして、前記画素データの各成分Ｉｒ，Ｉｇ，Ｉｂの各８bit を、各成分を示す複数のfloating型のデータｒ，ｇ，ｂへ変換する。

【００６１】（ｒ，ｇ，ｂはfloating型，０.０≦ｒ，ｇ，ｂ≦２５５.０）

前記の様に本発明のマイクロプロセッサにおける高速なunpack処理では、従来のunpack処理では必要としていたgeneral レジスタからfloatingレジスタへのデータ転送処理を行わずに、更に、前記実施例ではそのままfloatingレジスタで順次３回必要としていたbit 演算処理を１回のbit 演算処理で画素データのbit 演算処理を行い、integer 型のデータをfloating型のデータへ変換する。これにより、general レジスタからfloatingレジスタへのデータ転送処理が不要となり、更に、bit 演算処理を１回に削減しているので、unpack処理を著しく高速化することができる。

【００６２】以上、詳細に説明した様に、本発明のマイクロプロセッサは、グラフィックス・システムにおける画素データのpack処理，unpack処理において、floatingレジスタとgeneral レジスタ間のデータ転送処理を不要にし、更に、３回分のbit 演算処理を１回のbit 演算処理に削減している。これにより、本発明のマイクロプロセッサは、pack処理，unpack処理を著しく高速化することができる。

【００６３】又、本発明のマイクロプロセッサをＣＰＵに用いたグラフィックス・システムも、グラフィックス・システムにおける画素データのpack処理，unpack処理を著しく高速化することができる。

【００６４】

【発明の効果】以上、詳細に説明した様に、本発明のグラフィックス・システムに適したマイクロプロセッサ、及びそれを適用したグラフィックス・システムによれば、グラフィックス・システムにおける画素データに関するpack処理，unpack処理を著しく高速化することができるという特有の効果を奏する。

【図面の簡単な説明】

【図１】本発明のマイクロプロセッサのＦＵとＦＲｓの一実施例を示す構成図である。

【図２】本発明のマイクロプロセッサの一実施例を示す構成図である。

【図３】本発明のマイクロプロセッサの一実施例を示す構成図である。

【図４】従来のpack処理のアセンブリ言語によるプログラムとレジスタの状態を示す図である。

【図５】従来のunpack処理のアセンブリ言語によるプログラムとレジスタの状態を示す図である。

【図６】本発明のマイクロプロセッサにおける高速なpa

ck処理のアセンブリ言語によるプログラムとレジスタの状態を示す図である。

【図7】本発明のマイクロプロセッサにおける高速なunpack処理のアセンブリ言語によるプログラムとレジスタの状態を示す図である。

【図8】本発明のマイクロプロセッサにおける高速なpack処理のアセンブリ言語によるプログラムとレジスタの状態を示す図である。

【図9】本発明のマイクロプロセッサにおける高速なunpack処理のアセンブリ言語によるプログラムとレジスタの状態を示す図である。

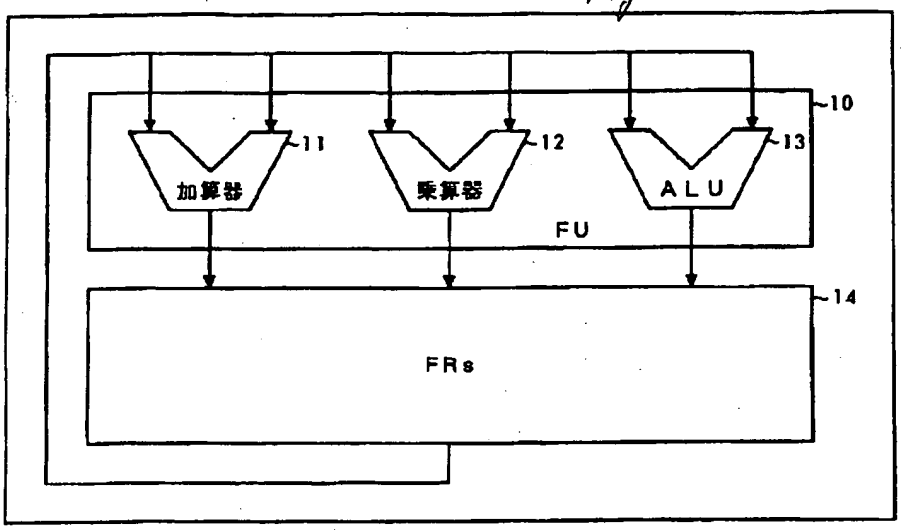
【図10】本発明のグラフィックス・システムの一実施例を示す構成図である。

【符号の説明】

10…FU、11…加算器、12…乗算器、13…ALU、14，27，311…FRs、21…System Interface、22…IC、23…DC、24…Controler、25…GRs、26…IU、28，310…FU、301，302，303…加算器、304，305，306…乗算器、307，308，309…ALU。

【図1】

Fig 1

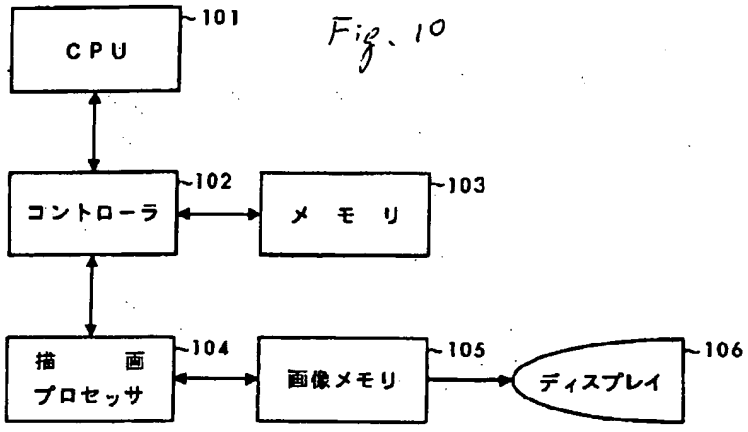


図一

【図10】

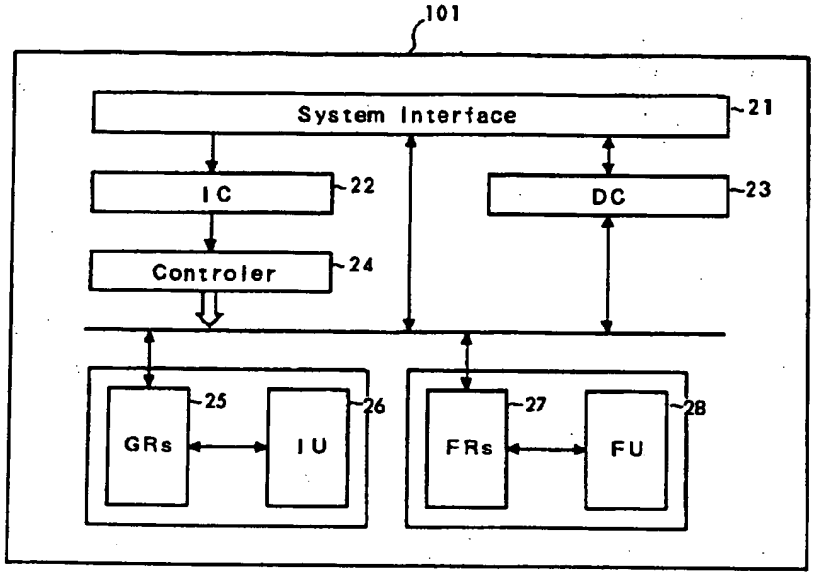
図 10

Fig. 10



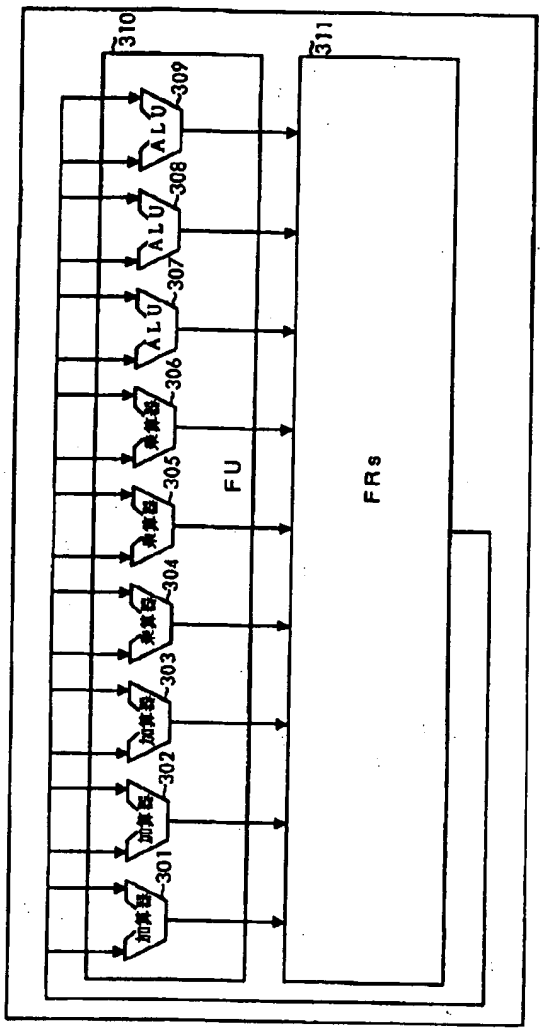
【図2】

図 2 FIG. 2


101
System Interface
21
IC
22
DC
23
Controler
24
GRs
25
IU
26
FRs
27
FU
28

【図3】 Fig. 3

図 3



【図7】 Fig 7

図 7

行番号	プログラム	レジスタの状態
L701	/* 初期状態：%fr19はIr,Ig,Ibの各8bitを	8bit 8bit 8bit 8bit ~702
L702	packした画素データ */	%fr19 = Ir Ig Ib
L703	/* %fr19をunpackし、	
L704	integer型からfloating型へ変換 */	32bit
L705	fextrucnvxf %fr19, 7,8,%fr16 /* %fr16 = r */	%fr16 = r ~705
L706	fextrucnvxf %fr19,15,8,%fr17 /* %fr17 = g */	%fr17 = g ~706
L707	fextrucnvxf %fr19,23,8,%fr18 /* %fr18 = b */	%fr18 = b ~707

【図4】 Fig. 4

図 4

行番号	プログラム	レジスタの状態
L401	/* 初期状態 */	
L402	/* %fr16 = r */	%fr16 = [r] (32bit) ~402
L403	/* %fr17 = g */	%fr17 = [g] (32bit) ~403
L404	/* %fr18 = b */	%fr18 = [b] (32bit) ~404
L405	/* floating型をinteger型へ変換 */	
L406	fcnvfx %fr16,%fr16 /* %fr16 = lr */	%fr16 = [8bit][8bit][8bit][lr] ~406
L407	fcnvfx %fr17,%fr17 /* %fr17 = lg */	%fr17 = [8bit][8bit][8bit][lg] ~407
L408	fcnvfx %fr18,%fr18 /* %fr18 = lb */	%fr18 = [8bit][8bit][8bit][lb] ~408
L409	/* floatingﾚｼﾞｽﾀからgeneralﾚｼﾞｽﾀへ転送 */	
L410	frnogr %fr16,%gr16 /* %gr16 = lr */	%gr16 = [8bit][8bit][8bit][lr] ~410
L411	frnogr %fr17,%gr17 /* %gr17 = lg */	%gr17 = [8bit][8bit][8bit][lg] ~411
L412	frnogr %fr18,%gr18 /* %gr18 = lb */	%gr18 = [8bit][8bit][8bit][lb] ~412
L413	/* lr,lg,lb各8bitを%gr19にpack処理する */	
L414	dep %gr16, 7,8,%gr19	%gr19 = [lr][8bit][8bit][8bit] ~414
L415	dep %gr17,15,8,%gr19	%gr19 = [lr][lg][8bit][8bit] ~415
L416	dep %gr18,23,8,%gr19	%gr19 = [lr][lg][lb][8bit] ~416

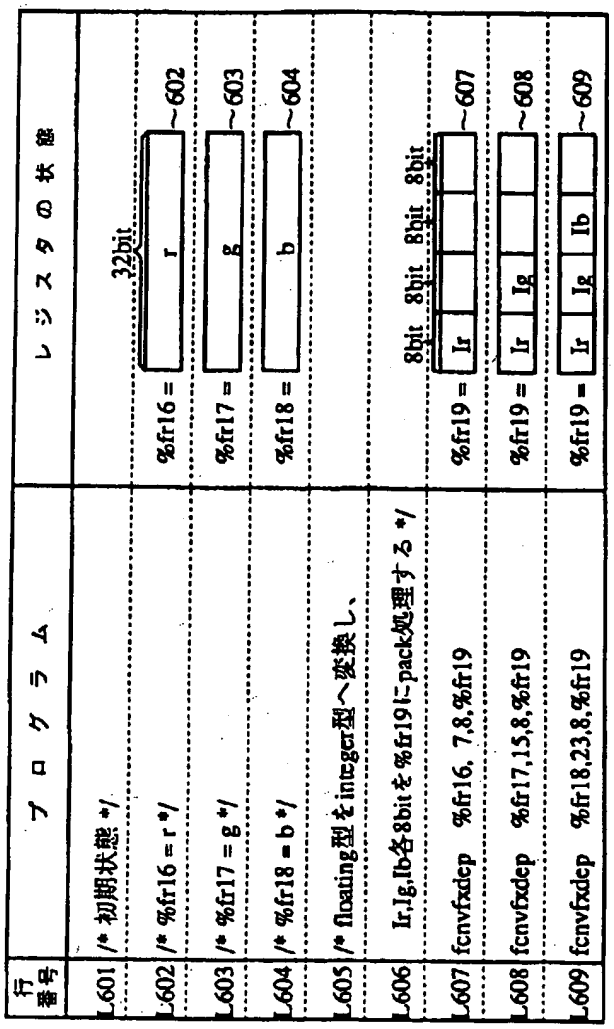
【図5】 Fig. 5

図 5

行番号	プログラム	レジスタの状態
L501	/* 初期状態：%gr19はIr,Ig,Ibの各8bitを	8bit 8bit 8bit 8bit
L502	packした画素データ */	%gr19 = Ir | Ig | Ib | ~502
L503	/* %gr19をunpack処理する */	
L504	extru %gr19, 7,8,%gr16 /* %gr16 = Ir */	%gr16 = | | | Ir ~504
L505	extru %gr19,15,8,%gr17 /* %gr17 = Ig */	%gr17 = | | | Ig ~505
L506	extru %gr19,23,8,%gr18 /* %gr18 = Ib */	%gr18 = | | | Ib ~506
L507	/* generalﾚｼﾞｽﾀからfloatingﾚｼﾞｽﾀへ転送 */	
L508	grtofr %gr16,%fr16 /* %fr16 = Ir */	%fr16 = | | | Ir ~508
L509	grtofr %gr17,%fr17 /* %fr17 = Ig */	%fr17 = | | | Ig ~509
L510	grtofr %gr18,%fr18 /* %fr18 = Ib */	%fr18 = | | | Ib ~510
L511	/* integer型をfloating型へ変換 */	32bit
L512	fcnvxf %fr16,%fr16 /* %fr16 = r */	%fr16 = r ~512
L513	fcnvxf %fr17,%fr17 /* %fr17 = g */	%fr17 = g ~513
L514	fcnvxf %fr18,%fr18 /* %fr18 = b */	%fr18 = b ~514

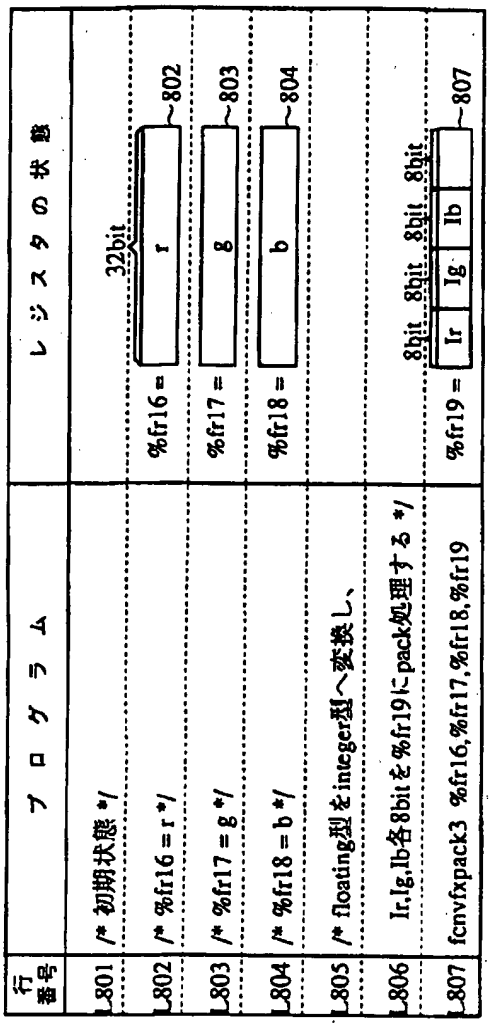
【図6】　Fig. 6

図 6


行番号
プログラム
レジスタの状態
L601 /* 初期状態 */
32bit
L602 /* %fr16 = r */
%fr16 =
r
~602
L603 /* %fr17 = g */
%fr17 =
g
~603
L604 /* %fr18 = b */
%fr18 =
b
~604
L605 /* floating型をinteger型へ変換し、
L606 Ir,Ig,Ibを各8bitを%fr19にpack処理する */
8bit 8bit 8bit 8bit
L607 fcnvfxdep %fr16, 7,8,%fr19
%fr19 =
Ir
~607
L608 fcnvfxdep %fr17,15,8,%fr19
%fr19 =
Ir Ig
~608
L609 fcnvfxdep %fr18,23,8,%fr19
%fr19 =
Ir Ig Ib
~609

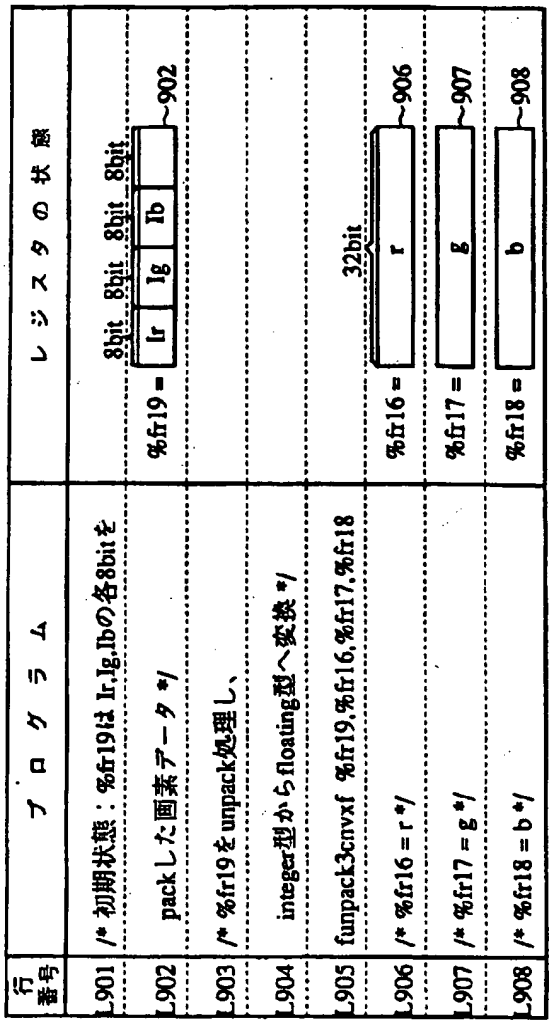
【図8】

図 8 Fig. 8



【図9】

図 9 Fig. 9



フロントページの続き

(72)発明者　藤井　秀樹
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所大みか工場内